Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 430949, 12 pages
http://dx.doi.org/10.1155/2014/430949

Research Article

Stochastic Constriction Cockroach Swarm Optimization for Multidimensional Space Function Problems

I. C. Obagbuwa, A. O. Adewumi, and A. A. Adebiyi

School Of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, Durban 4000, South Africa

Correspondence should be addressed to I. C. Obagbuwa; ibidunobagbuwa@yahoo.com

Received 27 November 2013; Accepted 6 January 2014; Published 16 March 2014

Academic Editor: Zhengguang Wu

The effect of stochastic constriction on cockroach swarm optimization (CSO) algorithm performance was examined in this paper. A stochastic constriction cockroach swarm optimization (SCCSO) algorithm is proposed. A stochastic constriction factor is introduced into CSO algorithm for swarm stability enhancement; control cockroach movement from one position to another while searching for solution to avoid explosion; enhanced local and global searching capabilities. SCCSO performance was tested through simulation studies and its performance on multidimensional functions is compared with that of original CSO, modified cockroach swarm optimization (MCSO), and one of the well-known global optimization techniques in the literature known as line search restart techniques (LSRS). Standard benchmarks that have been widely used for global optimization problems are considered for evaluating the proposed algorithm. The selected benchmarks were solved up to 3000 dimensions by the proposed algorithm.

1. Introduction

Swarm intelligence (SI) models collective social animal behaviour. Cockroach optimization is a new development under SI paradigm; cockroach optimization algorithms [1–3] are inspired by collective cockroach social behaviour.

Obtaining accurate and efficient optimization algorithms with good speed is the desire of optimization research community. Global optimization algorithms have been improved upon in the literature with different techniques to be able to solve high dimension problems. Examples include an improved particle swarm optimization (IPSO) [4] that was used to evaluate problems with 100 and 150 dimensions; parallel particle swarm approach [5] that solves problems up to 128 dimensions; particle swarm optimization (PSO), genetic algorithm (GA), and line search restart (LSRS) approaches were used in [6] for high dimension problems; PSO, GA, and LSRS performances were compared from 50 to 1000 dimensions; LSRS performs better than PSO and GA. LSRS was tested further for 2000 dimensions [6].

Stochastic constriction is used in this paper to improve the performance of original cockroach swarm optimization algorithm (CSO) [2]. A stochastic constriction cockroach swarm optimization (SCCSO) algorithm is presented. The constriction factor controls cockroach movement and prevents swarm explosion; cockroach is able to exploit local neighbourhood and explore the search space.

The performance of SCCSO is investigated and compared with that of original CSO, modified cockroach swarm optimization (MSCO) [3], and LSRS [6] on high dimension problems for finding the global optimal. LSRS was considered for performance comparison with the proposed algorithm because it has been proved for high dimension in the literature [6].

The selected benchmarks are multimodal and multidimensional problems which are considered as very hard global optimization problems. The global optimization problems of the form: minimize $f(x)$ subject to $x \in \Omega$, where x is a continuous variable vector with domain $\Omega \subset \mathfrak{R}^D$ and $f(x) : \Omega \rightarrow \mathfrak{R}$ is a continuous real-valued function. Between the upper and lower limits of each dimension, Ω is described. x^* represents global solution, while $f(x^*)$ represents the corresponding function fitness value [7]. The problems are described in Table 1 of this paper.

The effects of stochastic constriction factor on cockroach swarm optimization (CSO) algorithm improve its accuracy, and the algorithm solves multidimensional benchmark problems to 3000 dimensions. The organization of the remaining

Table 1: Benchmark functions.

Problems	Range	Minimum
Ackley: $f(x) = +20 + e - 20e^{-0.2\sqrt{\sum_{i=1}^{n}(x_i^2/n)}} - e^{\sum_{i=1}^{n}\cos(2\pi x_i/n)}$	$[-10, 10]$	0
Levy: $f(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_i + 1)] + (y_n - 1)^2(1 + \sin^2(2\pi x_n))$, where $y_i = 1 + \frac{1}{4}(x_i - 1)$	$[-10, 10]$	0
Quadric: $f(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-10, 10]$	0
Rastrigin: $f(x) = 10n + \sum_{i=1}^{n}\frac{x_i^{2}}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-5.12, 5.12]$	0
Rosenbrock: $f(x) = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]$	$[-5, 10]$	0
Schwefel: $f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-10, 10]$	0
Sphere: $f(x) = \sum_{i=1}^{n} x_i^2$	$[-10, 10]$	0
Sum Squares: $f(x) = \sum_{i=1}^{n} i x_i^2$	$[-10, 10]$	0
Griewank: $f(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0

part of this paper is as follows. Section 2 presents the existing CSO, MCSO, and the proposed SCCSO models; Section 3 presents the simulation studies with the obtained results; Section 4 discusses the results of experiments; and the paper is summarised in Section 5.

2. Cockroach Swarm Optimization

CSO algorithm is a population based global optimization algorithm; it was introduced by ZhaoHui and HaiYan [2] and was modified by ZhaoHui [3] with the introduction of inertial weight. CSO has been applied to problems in the literature [8, 9]. CSO models mimic cockroach behaviours which are chase-swarming, dispersing, and ruthless behaviours. CSO [2] models are given as follows.

(1) Chase-swarming behaviour:

$$x_i = \begin{cases} x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i), & x_i \neq p_i \\ x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i), & x_i = p_i, \end{cases} \quad (1)$$

where x_i is the cockroach position, step is a fixed value, rand is a random number within $[0, 1]$, p_i is the personal best position, and p_g is the global best position.

Consider

$$p_i = \text{Opt}_j\left\{x_j, \left|x_i - x_j\right| \leq \text{visual}\right\}, \quad (2)$$

where visual perception distance is a constant. $j = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, N$

$$p_g = \text{Opt}_i\left\{x_i\right\}. \quad (3)$$

(2) Dispersion behaviour

$$x_i = x_i + \text{rand}(1, D), \quad i = 1, 2, \ldots, N, \quad (4)$$

where $\text{rand}(1, D)$ is a D-dimensional random vector that can be set within a certain range.

(3) Ruthless behaviour

$$x_k = p_g, \quad (5)$$

where k is a random integer within $[1, N]$ and p_g is the global best position.

2.1. Modified Cockroach Swarm Optimization. MCSO [3] extends CSO [2] with the introduction of inertial weight in chase-swarming behaviour as shown below. Other models remain the same as in CSO [2].

Chase-swarming behaviour

$$x_i = \begin{cases} w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i), & x_i \neq p_i \\ w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i), & x_i = p_i, \end{cases} \quad (6)$$

where w is an inertial weight which is a constant.

2.2. Stochastic Constriction Cockroach Swarm Optimization. Constriction factor was introduced by Clerc and Kennedy [10] to prevent swarm explosion in particle swarm optimization (PSO); PSO is one of the existing and popular SI algorithms. They proposed a constriction as

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \quad (7)$$

where $\varphi = c_1 + c_2$, $\varphi > 4.0$, $c1$ is the recognition factor, and $c2$ is the social factor.

Shi described χ as a constant which is approximately 0.729; φ is commonly set to 4.1 [11]. An algorithm with constriction constant 0.729 is equivalent to algorithm of inertia weight 0.729 with $c1 = c2 = 1.49445$ [11].

```
INPUT: Fitness function: f(x), x ∈ R^D
set parameters and generate an initial population of cockroach
set p_g = x_1
for i = 2 to N do
    if f(x_i) < f(p_g) then
      p_g = x_i
    end if
end for
for t = 1 to T_max do
    for i = 1 to N do
        for j = 1 to N do
            if abs(x_i − x_j) < visual; f(x_j) < f(x_i) then
                p_i = x_i
            end if
        end for
        if p_i == x_i then
            x_i = ξ (x_i + step · rand · (p_g − x_i))
        else
            x_i = ξ (x_i + step · rand · (p_i − x_i))
        end if
        if f(x_i) < f(p_g) then
            p_g = x_i
        end if
    end for
    for i = 1 to N do
        x_i = x_i + rand (1, D)
        if f(x_i) < f(p_g) then
            p_g = x_i
        end if
    end for
    k = randint ([1, N])
    x_k = p_g;
end for
Check termination condition
```

Algorithm 1: Stochastic constriction cockroach swarm optimization algorithm.

Constriction PSO was experimentally compared with inertia weight PSO [12, 13]; constriction PSO performs better than inertia weight PSO.

Similarly, a constriction factor is considered in this paper to control cockroach movement during swarming process for avoidance of swarm explosion. We use a stochastic constriction factor (SCF) instead of a constant constriction. SCF allows generation of different values as constriction factor in each iteration.

SCF helps to maintain the stability of swarm, enhances local and global search, and improves the speed and convergence of the algorithm. The algorithm utilized little cpu time in seconds to solve multidimensional problems and obtain optimal results. SCF was investigated through simulation studies; experiments and results are presented in Section 3.

The chase-swarming behaviour, (1) of CSO [2] is modified in this paper with the introduction of SCF ξ. ξ randomly takes values between zero and one in each iteration. ξ controls entire cockroach movement, not only cockroach position as with inertial weight w of CSO in [3]. Chase-swarming equation now becomes

$$x_i = \begin{cases} \xi\left(x_i + \text{step} \cdot \text{rand} \cdot \left(p_i - x_i\right)\right), & x_i \neq p_i \\ \xi\left(x_i + \text{step} \cdot \text{rand} \cdot \left(p_g - x_i\right)\right), & x_i = p_i. \end{cases} \tag{8}$$

The algorithmic steps for SCCSO are illustrated in Algorithm 1 and its computational steps are given as follows.

(1) Initialise cockroach swarm with uniform distributed random numbers and set all parameters with values.

(2) Find p_i and p_g using (2) and (3).

(3) Exhibit chase-swarming using (8).

(4) Exhibit dispersion behaviour using (4).

(5) Exhibit ruthless behaviour using (5).

(6) Repeat the loop until stopping criteria is reached.

Table 2: Perfomance of CSO, MCSO, and SCCSO for dimension 10.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square
CSO									
Best	$4.01E-04$	$1.66E-07$	$1.96E-09$	$6.98E-06$	$5.40E-04$	$2.57E-07$	$6.33E-06$	$1.70E-05$	$1.58E-04$
Average	9.23	$8.07E+01$	$1.85E-04$	$1.99E-01$	$1.11E+06$	$2.56E-04$	$2.45E-01$	$2.44E-01$	$7.35E02$
STD	5.45	$2.11E+02$	$2.75E-04$	$8.90E-01$	$2.13E+06$	$2.89E-04$	$3.18E-04$	$7.09E-01$	$1.64E03$
Success	1/20	13/20	20/20	19/20	1/20	20/20	20/20	17/20	6/20
Time	45.42	32.96	2.26	14.81	44.31	5.99	4.0	20.37	35.30
MCSO									
Best	$4.37E-10$	1.38	$1.20E-21$	0.00	9.00	$1.59E-21$	$9.90E-23$	0.00	$1.97E-17$
Average	$5.16E-07$	1.38	$5.21E-14$	$1.85E-12$	9.00	$1.41E-12$	$2.10E-14$	$8.75E-11$	$1.59E-12$
STD	$1.11E-06$	$2.28E-16$	$1.58E-13$	$6.72E-12$	0.00	$6.26E-12$	$9.07E-14$	$3.77E-10$	$4.09E-12$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.21	49.73	0.12	0.14	37.01	0.143	0.12	0.15	0.12
SCCSO									
Best	$8.88E-16$	1.38	$2.62E-55$	0.00	9.00	$1.60E-59$	$8.10E-52$	0.00	$1.21E-50$
Average	$1.07E-15$	1.38	$1.50E-33$	0.00	9.00	$5.15E-30$	$8.10E-26$	0.00	$6.81E-27$
STD	$7.94E-16$	$2.28E-16$	$6.64E-33$	0.00	0.00	$2.30E-29$	$3.62E-25$	0.00	$3.04E-26$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.13	48.65	0.12	0.14	36.37	0.13	0.12	0.14	0.12

STD denotes standard deviation. Time denotes execution time in seconds.

The performance of the proposed algorithm on well-known high dimension benchmarks is evaluated in Section 3.

3. Simulation Studies

Experiments were conducted in 2 stages to investigate the speed, accuracy, robustness, stability, and searching capabilities of SCCSO. Its performance is compared with that of existing CSO and MCSO algorithm [2, 3] and LSRS approach [6] for high dimension. Table 1 of this paper shows the considered benchmarks for experiments which was adopted from the literature [2, 3, 6]. These are a set of standard benchmarks that have the diverse characteristics which can cover various global optimization problems. SCCSO, CSO, and MCSO algorithms were implemented in MATLAB 7.14 (R2012a) and run on a system with 2.30 GHz processor with 4.00 GB of RAM.

The numerical results for SCCSO, CSO, and MCSO were obtained after implementing the algorithms in this paper, but the numerical results for LSRS were as reported [6].

Experiment parameters of [2, 3] is used in this paper, perception distance visual = 5; the largest step size step = 2; experiments run 20 times with 1000 maximum iteration each; and we use swarm size 50. Global minimum values of each benchmark functions were evaluated in each experiment; best and average performance and standard deviation of the average optimal values were recorded during experiments.

3.1. Stage One Experiments and Results. Investigation was done in this stage on the performance and comparison of the SCCSO, CSO, and MCSO for dimensions 10, 20, 30, and 40. Inertial weight $w = 0.618$ was adopted from [3] for MCSO. Success rate (SRT) and computation time in seconds were recorded during experiments.

SRT of an algorithm is the number of successful runs out of predefined number of runs. A run is considered successful when algorithm finds optimal solution, when the algorithm converges to a good solution and the function value satisfies $|f(x^*) - f_{\min}| \leq \epsilon$ [14]. ϵ is the desired accuracy, $f(x^*)$ is the optimum fitness value, and $f_{\min}$ is the global minimum value. SRT = SR/TR, SR denotes number of successful runs, and TR denotes total number of runs [14].

SCCSO and MSCO algorithms have 100% SRT on benchmarks except Levy and Rosenbrock. Table 9 depicts the comparison results of SRT of CSO, MCSO, and SCCSO algorithms; CSO has low SRT for most of the benchmarks.

The computation time for each benchmark in the experiments with CSO, MCSO, and SCCSO is depicted in Table 10; more CPU time is utilized by CSO algorithm than MCSO and SCCSO algorithms in evaluating the benchmarks for dimension 10, 20, 30, and 40. SCCSO utilized minimum time in solving the selected test functions.

The performance of SCCSO, CSO, and MCSO for 10, 20, 30, and 40 dimensions, respectively, is shown in Tables 2, 3, 4, and 5 of this paper. Comparison of best performance depicted in Table 6 clearly shows that SCCSO performs better than CSO and MCSO. Table 7 gives a comparison of average performance of SCCSO, CSO, and MCSO; SCCSO algorithm performs better than others. Similarly, the comparison results of standard deviation of the average optimal shown in Table 8 reveal that SCCSO has better standard deviation than others. Bold values indicated better minimum optimal values, and "—" in Tables 7 and 8 indicated no good optimal value.

3.2. Stage Two Experiments and Results. Stage two experiments investigate the performance of the proposed algorithm

Table 3: Perfomance of CSO, MCSO, and SCCSO for dimension 20.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square
CSO									
Best	2.00E01	3.38$E-05$	6.13$E-10$	1.52$E-05$	1.61E06	3.61$E-06$	7.00$E-08$	1.86$E-04$	1.58$E-04$
Average	2.14E01	1.27E04	2.61$E-04$	8.46E03	1.08E11	3.74$E-04$	2.43E03	9.22	1.68E06
STD	8.75$E-01$	1.79E04	2.16$E-04$	1.52$E-05$	2.36E11	3.59$E-04$	1.08E04	1.22E01	4.40E06
Success	0/20	3/20	20/20	8/20	0/20	20/20	18/20	4/20	1/20
Time	63.95	71.33	18.20	49.34	61.48	16.19	19.06	57.47	57.37
MCSO									
Best	1.12$E-08$	2.29	1.4881$E-20$	0.00	1.90E01	9.67$E-22$	7.29$E-19$	0.00	1.70$E-15$
Average	1.02$E-06$	2.29	1.16$E-13$	2.80$E-11$	1.90E01	2.46$E-15$	3.30$E-14$	2.27$E-10$	3.34$E-11$
STD	1.17$E-06$	9.11$E-16$	4.17$E-13$	9.55$E-11$	0.00	6.06$E-15$	1.03$E-13$	1.01$E-09$	1.00$E-10$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.21	73.729	0.21	0.20	54.506	0.17	0.18	0.18	0.16
SCCSO									
Best	8.88$E-16$	2.29	5.25$E-49$	0.00	1.90E01	7.32$E-58$	2.13$E-55$	0.00	6.17$E-47$
Average	8.88$E-16$	2.29	1.55$E-29$	0.00	1.90E01	7.66$E-35$	6.48$E-29$	0.00	1.19$E-31$
STD	0.00	9.11$E-16$	4.77$E-29$	0.00	0.00	2.25$E-34$	2.90$E-28$	0.00	4.12$E-31$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.18	74.22	0.20	0.18	54.59	0.206	0.17	0.17	0.21

STD denotes standard deviation. Time denotes execution time in seconds.

Table 4: Perfomance of CSO, MCSO, and SCCSO for dimension 30.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square
CSO									
Best	1.06	6.40$E-04$	9.54$E-06$	6.01$E-05$	4.77E02	6.75$E-08$	7.14$E-10$	1.41$E-04$	2.11$E-05$
Average	2.01	3.67E05	2.03E03	1.00E04	9.22E11	3.28	2.61$E-04$	2.19E01	4.49E06
STD	4.71	1.15E06	8.96E03	3.10E04	3.15E12	1.47E01	3.25$E-04$	3.10E01	1.65E07
Success	0/20	1/20	18/20	5/20	0/20	19/20	20/20	2/20	1/20
Time	81.17	112.78	29.03	66.77	79.45	27.03	24.49	75.20	75.62
MCSO									
Best	5.05$E-12$	3.20	1.05$E-20$	0.00	2.90E01	8.52$E-23$	7.29$E-21$	0.00	1.30$E-17$
Average	8.31$E-06$	3.20	7.40$E-12$	1.16$E-12$	2.90E01	7.80$E-13$	2.77$E-13$	2.14$E-12$	5.54$E-11$
STD	3.0$E-05$	4.56$E-16$	3.31$E-11$	3.04$E-12$	0.00	3.40$E-12$	8.28$E-13$	8.05$E-12$	1.19$E-10$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.23	98.69	0.19	0.21	72.34	0.21	0.19	0.226	0.22
SCCSO									
Best	8.88$E-16$	3.20	5.77$E-57$	0.00	2.90E01	1.09$E-49$	4.55$E-53$	0.00	3.04$E-49$
Average	5.64$E-13$	3.20	1.34$E-30$	0.00	2.90E01	7.10$E-30$	6.54$E-36$	0.00	2.28$E-29$
STD	2.51$E-12$	4.55$E-16$	5.86$E-30$	0.00	0.00	3.15$E-29$	2.57$E-35$	0.00	9.90$E-29$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.23	98.59	0.24	0.24	73.55	0.20	0.24	0.31	0.20

STD denotes standard deviation. Time denotes execution time in seconds.

SCCSO for dimensions 50, 100, 500, 1000, 2000, and 3000. SCCSO solves benchmarks to minimum values: for instance, the algorithm finds optimal value for Rastrigin function.

The performance of SCCSO is compared with that of LSRS which has been tested for dimensions 50, 100, 500, 1000, and 2000 [6]. The best and average performance, and standard deviation of average optimum values of SCCSO and LSRS is shown in Tables 11, 12, 13, 14, and 15 respectively. Performance of SCCSO for 3000 dimensions is shown in Table 16.

To determine the significant difference of SCCSO and LSRS average performance, test statistics of analysis of variance (ANOVA) was used. Average performance of SCCSO and LSRS is shown in Table 17; and the results of ANOVA test is in Table 18 with the P values shown in the last column of the table. P value for Ackley is 0.309; Quadric is 0.346,

Table 5: Perfomance of CSO, MCSO, and SCCSO for dimension 40.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square
CSO									
Best	1.97E01	2.26$E-05$	8.12$E-07$	2.50$E-07$	6.88E02	3.83$E-07$	2.27$E-07$	1.24$E-05$	1.74E02
Average	2.15E01	4.96E04	6.81E01	2.70E03	4.17E13	1.92E01	3.25$E-04$	1.87E01	3.17E06
STD	7.77$E-01$	1.35E05	2.10$E-02$	5.26E03	1.84E14	8.60E01	2.90$E-04$	2.26E01	3.66E06
Success	0/20	1/20	18/20	6/20	0/20	19/20	20/20	5/20	0/20
Time	100.53	127.913	39.61	79.75	98.02	34.90	20.30	88.27	95.44
MCSO									
Best	1.21$E-10$	4.11	1.64$E-21$	0.00	3.90E01	6.75$E-22$	2.29$E-23$	0.00	5.15$E-15$
Average	3.05$E-06$	4.11	1.89$E-13$	2.85$E-10$	3.90E01	1.53$E-13$	2.99$E-12$	1.57$E-12$	1.71$E-09$
STD	1.04$E-05$	9.11$E-16$	8.31$E-13$	9.95$E-10$	0.00	5.03$E-13$	1.32$E-11$	3.85$E-12$	6.78$E-09$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.26	123.833	0.26	0.27	90.43	0.24	0.25	0.24	0.24
SCCSO									
Best	8.88$E-16$	4.11	2.8806$E-55$	0.00	3.90E01	4.56$E-54$	1.40$E-50$	0.00	9.19$E-48$
Average	2.49$E-15$	4.11	1.56$E-32$	0.00	3.90E01	1.73$E-34$	8.76$E-31$	0.00	1.56$E-24$
STD	6.36$E-15$	9.11$E-16$	6.96$E-32$	0.00	0.00	6.99$E-34$	3.91$E-30$	0.00	6.98$E-24$
Success	20/20	0/20	20/20	20/20	0/20	20/20	20/20	20/20	20/20
Time	0.25	123.04	0.24	0.25	90.99	0.25	0.27	0.28	0.31

STD denotes standard deviation. Time denotes execution time in seconds.

Table 6: Comparison of best perfomance of CSO, MCSO, and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square	Number of good optimal
Dim 10										
CSO	4.01$E-04$	**1.66E − 07**	1.93$E-09$	6.97$E-06$	**5.40E − 04**	2.57$E-07$	6.33$E-06$	1.79$E-05$	1.56$E-04$	2
MCSO	4.37$E-10$	1.38	1.20$E-21$	**0.00**	9.00	1.59$E-21$	9.90$E-23$	**0.00**	1.97$E-17$	2
SCCSO	**8.88E − 16**	1.38	**2.62E − 55**	**0.00**	9.00	**1.60E − 59**	**8.11E − 52**	**0.00**	**1.21E − 50**	7
Dim 20										
CSO	2.00E01	**3.38E − 05**	6.13$E-10$	1.52$E-05$	1.61E06	3.61$E-06$	7.00$E-08$	1.86$E-04$	1.58$E-04$	1
MCSO	1.12$E-08$	2.29	1.49$E-20$	**0.00**	**1.90E01**	9.67$E-22$	7.29$E-19$	**0.00**	1.709$E-15$	3
SCCSO	**8.88E − 16**	2.29	**5.25E − 49**	**0.00**	**1.90E01**	**7.32E − 58**	**2.15E − 55**	**0.00**	**6.17E − 47**	8
Dim 30										
CSO	1.06	**6.41E − 04**	9.54$E-06$	6.01$E-05$	4.77E02	6.75$E-08$	7.14$E-10$	1.41$E-04$	2.11$E-05$	1
MCSO	5.05$E-12$	3.20	1.05$E-20$	**0.00**	**2.90E01**	8.52$E-23$	7.29$E-21$	**0.00**	1.30$E-17$	3
SCCSO	**8.88E − 16**	3.20	**5.77E − 57**	**0.00**	**2.90E01**	**1.09E − 49**	**4.559E − 53**	**0.00**	**3.04E − 49**	8
Dim 40										
CSO	1.97E01	**2.26E − 05**	8.12$E-07$	2.50$E-07$	6.88E02	3.83$E-07$	2.23$E-07$	1.24$E-05$	1.74E02	1
MCSO	1.21$E-10$	4.11	1.64$E-21$	**0.00**	**3.90E01**	6.75$E-22$	2.29$E-23$	**0.00**	5.15$E-15$	3
SCCSO	**8.88E − 16**	4.11	**2.88E − 55**	**0.00**	**3.90E01**	**4.56E − 54**	**1.40E − 50**	**0.00**	**9.19E − 48**	8

Total number of good optimum (best): CSO (5); MCSO (11); SCCSO (31).

Levy is 0.076; Rosenbrock is 0.078; Schwefel is 0.320; Sphere is 0.145; Sum square is 0.345; and Rastrigin P value cannot be determined because both LSRS and SCCSO have same optimal values. P values for all tested functions are greater than the literature threshold value of 0.05. This means, there is no significant difference in the average performance of LSRS and SCCSO for the selected test problems. Figure 1 show graphical illustration of the ANOVA test.

4. Discussions

Stages one and two experiments investigated the performance of the proposed algorithm and compared its performance with existing algorithms for high dimension. Stage one compared three types CSO based algorithms: CSO with no inertial weight [2]; CSO with inertial weight [3]; and the proposed CSO with stochastic constriction factor. The

Table 7: Comparison of average perfomance of CSO, MCSO, and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square	Number of good optimal
Dim 10										
CSO	9.23	$8.07E+01$	$1.85E-04$	$1.99E-01$	$1.11E+06$	$2.56E-04$	$2.45E-01$	$2.44E-01$	$7.35E02$	—
MCSO	$5.16E-07$	**1.38**	$5.21E-14$	$1.85E-12$	**9.00**	$1.41E-12$	$2.10E-14$	$8.75E-11$	$1.59E-12$	2
SCCSO	**1.07E – 15**	**1.38**	**1.50E-33**	**0.00**	**9.00**	**5.15E – 30**	**8.10E – 26**	**0.00**	**6.81E – 27**	9
Dim 20										
CSO	$2.14E01$	$1.27E04$	$2.61E-04$	$8.46E03$	$1.08E11$	$3.74E-04$	$2.43E03$	9.22	$1.68E06$	—
MCSO	$1.02E-06$	**2.29**	$1.16E-13$	$2.80E-11$	**1.90E01**	$2.46E-15$	$3.30E-14$	$2.27E-10$	$3.34E-11$	2
SCCSO	**8.88E – 16**	**2.29**	**1.55E – 29**	**0.00**	**1.90E01**	**7.66E – 35**	**6.48E – 29**	**0.00**	**1.19E – 31**	9
Dim 30										
CSO	2.01	$3.67E05$	$2.03E03$	$1.00E04$	$9.22E11$	3.28	$2.61E-04$	$2.19E01$	$4.49E06$	—
MCSO	$8.31E-06$	**3.20**	$7.40E-12$	$1.16E-12$	**2.90E01**	$7.80E-13$	$2.77E-13$	$2.14E-12$	$5.54E-11$	2
SCCSO	**5.64E – 13**	**3.20**	**1.34E – 30**	**0.00**	**2.90E01**	**7.10E – 30**	**6.54E – 36**	**0.00**	**2.28E – 29**	9
Dim 40										
CSO	$2.15E01$	$4.96E04$	$6.81E01$	$2.70E03$	$4.17E13$	$1.92E01$	$3.25E-04$	$1.87E01$	$3.17E06$	—
MCSO	$3.05E-06$	**4.11**	$1.89E-13$	$2.85E-10$	**3.90E01**	$1.53E-13$	$2.99E-12$	$1.57E-12$	$1.71E-09$	2
SCCSO	**2.49E – 15**	**4.11**	**1.56E – 32**	**0.00**	**3.90E01**	**1.73E – 34**	**8.76E – 31**	**0.00**	**1.56E – 24**	9

Total number of good optimum (average): CSO (0); MCSO (8); SCCSO (36).

Table 8: Comparison of standard deviation of mean optimal for CSO, MCSO, and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square	Number of good STD
Dim 10										
CSO	5.45	$2.11E+02$	$2.75E-04$	$8.90E-01$	$2.13E+06$	$2.89E-04$	$3.18E-04$	$7.09E-01$	$1.64E03$	—
MCSO	$1.11E-06$	**2.28E – 16**	$1.58E-13$	$6.72E-12$	**0.00**	$6.26E-12$	$9.07E-14$	$3.77E-10$	$4.09E-12$	2
SCCSO	**7.94E – 16**	**2.28E – 16**	**6.64E – 33**	**0.00**	**0.00**	**2.30E – 29**	**3.62E – 25**	**0.00**	**3.04E – 26**	9
Dim 20										
CSO	$8.75E-01$	$1.79E04$	$2.16E-04$	$1.52E-05$	$2.36E11$	$3.59E-04$	$1.08E04$	$1.22E01$	$4.40E06$	—
MCSO	$1.17E-06$	**9.11E – 16**	$4.17E-13$	$9.55E-11$	**0.00**	$6.06E-15$	$1.03E-13$	$1.01E-09$	$1.00E-10$	2
SCCSO	**0.00**	**9.11E – 16**	**4.77E – 29**	**0.00**	**0.00**	**2.25E – 34**	**2.90E – 28**	**0.00**	**4.12E – 31**	9
Dim 30										
CSO	4.71	$1.15E06$	$8.96E03$	$3.10E04$	$3.15E12$	$1.47E01$	$3.25E-04$	$3.10E01$	$1.65E07$	—
MCSO	$3.0E-05$	**4.56E – 16**	3.31E – 11	3.04E – 12	**0.00**	$3.40E-12$	$8.28E-13$	$8.05E-12$	$1.19E-10$	2
SCCSO	**2.51E – 12**	**4.55E – 16**	**5.86E – 30**	**0.00**	**0.00**	**3.15E – 29**	**2.57E – 35**	**0.00**	**9.90E – 29**	9
Dim 40										
CSO	$7.77E-01$	$1.35E05$	$2.10E-02$	$5.26E03$	$1.84E14$	$8.60E01$	$2.90E-04$	$2.26E01$	$3.66E06$	—
MCSO	$1.04E-05$	**9.11E – 16**	$8.31E-13$	$9.95E-10$	**0.00**	$5.03E-13$	$1.32E-11$	$3.85E-12$	$6.78E-09$	2
SCCSO	**6.36E – 15**	**9.11E – 16**	**6.96E – 32**	**0.00**	**0.00**	**6.99E – 34**	**3.91E – 30**	**0.00**	**6.98E – 24**	9

Total number of good standard deviations: CSO (0); MCSO (8); SCCSO (36).

stochastic constriction factor enhances the performance of proposed algorithm for solving high dimension problems as clearly shown in the results of experiments conducted.

The comparison results of stage one experiments for dimensions 10, 20, 30, and 40 clearly show that proposed algorithm outperforms CSO and MCSO. The proposed algorithm results show better best optimal results, better average optimal results and better standard deviation of mean optimal, and better execution time than CSO and MCSO in Tables 6, 7, 8, and 10, respectively. Both MCSO and SCCSO have similar SRT as shown in Table 9.

SCCSO has been shown to be better than CSO and MCSO for high dimension problems to 40 dimension in the experiments conducted in this paper; it was investigated further on benchmarks up to 3000 dimensions. SCCSO performance was compared with that of LSRS in stage two experiments for 50, 100, 500, 1000, and 2000 dimensions. The statistical analysis conducted on the results revealed that

Table 9: Comparison of success rate of CSO, MCSO, and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square	Number of 100% success rate
Dim 10										
CSO	0.05	0.65	1	0.95	0.05	1	1	0.85	0.3	3
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 20										
CSO	0	0.15	1	0.4	0	1	0.9	0.2	0.05	2
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 30										
CSO	0	0.05	0.9	0.25	0	0.95	1	0.1	0.1	1
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7
Dim 40										
CSO	0	0.05	0.9	0.3	0	0.95	1	0.25	0	1
MCSO	1	0	1	1	0	1	1	1	1	7
SCCSO	1	0	1	1	0	1	1	1	1	7

Total number of good success rate: CSO (7); MCSO (28); SCCSO (28).

Table 10: Comparison of execution time (in seconds) for CSO, MCSO, and SCCSO.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Griewangk	Sum Square	Number of good time
Dim 10										
CSO	45.42	**32.96**	2.26	14.81	44.31	5.99	4.0	20.37	35.30	1
MCSO	0.21	49.73	**0.12**	**0.14**	37.05	0.14	**0.12**	0.15	**0.12**	4
SCCSO	**0.13**	48.65	**0.12**	**0.14**	**36.37**	**0.13**	**0.12**	**0.14**	**0.12**	8
Dim 20										
CSO	63.95	**71.33**	18.20	49.34	61.48	16.19	19.06	57.47	57.37	1
MCSO	0.21	73.73	0.21	0.20	**54.51**	**0.17**	0.18	0.18	**0.16**	3
SCCSO	**0.18**	74.22	**0.20**	**0.18**	54.59	0.21	**0.17**	**0.17**	0.21	5
Dim 30										
CSO	81.17	112.78	29.03	66.77	79.45	27.03	24.49	75.20	75.62	—
MCSO	**0.23**	98.69	**0.19**	**0.21**	**72.34**	0.21	**0.19**	**0.23**	0.22	6
SCCSO	**0.23**	**98.59**	0.24	0.24	73.55	**0.20**	0.24	0.31	**0.20**	4
Dim 40										
CSO	100.53	127.91	39.61	79.75	98.02	34.90	20.30	88.27	95.44	—
MCSO	0.26	123.83	0.26	0.27	**90.43**	**0.24**	**0.25**	**0.24**	**0.24**	5
SCCSO	**0.25**	**123.04**	**0.24**	**0.25**	90.99	0.25	0.27	0.28	0.31	4

Total number of good execution time: CSO (2); MCSO (18); SCCSO (21).

Table 11: Perfomance of LSRS and SCCSO for dimension 50.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
LSRS								
Best	$-6.5E-19$	$2.9E-39$	$6.27E-19$	0.00	$2.47E-28$	$1.86E-11$	$1.34E-22$	$9.86E-21$
Average	$-6.5E-19$	$2.9E-39$	$2.33E-18$	0.00	$1.38E-18$	$1.91E-11$	$1.38E-18$	$1.42E-18$
STD	0.00	0.00	$8.11E-19$	0.00	$1.29E-18$	$4.15E-12$	$1.29E-18$	$1.25E-18$
SCCSO								
Best	$8.88E-16$	5.01	$5.03E-54$	0.00	$4.90E01$	$5.61E-51$	$1.47E-51$	$4.43E-45$
Average	$8.88E-16$	5.01	$7.25E-31$	0.00	$4.90E01$	$1.02E-28$	$2.62E-28$	$4.54E-28$
STD	0.00	$9.15E-16$	$3.06E-30$	0.00	0.00	$4.54E-28$	$1.17E-27$	$1.98E-27$

Table 12: Perfomance of LSRS and SCCSO for dimension 100.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
LSRS								
Best	$-6.5E-19$	$2.9E-39$	$9.20E-16$	0.00	$5.83E-28$	$7.81E-19$	$5.34E-19$	$4.68E-18$
Average	$-6.5E-19$	$2.9E-39$	$1.15E-15$	0.00	$6.94E-16$	$3.98E-10$	$6.94E-16$	$6.98E-16$
STD	0.00	0.00	4.38E-16	0.00	$6.63E-16$	$4.97E-10$	$6.63E-16$	$6.58E-16$
SCCSO								
Best	$8.88E-16$	9.56	$1.40E-54$	0.00	$9.90E01$	$1.43E-51$	$1.93E-49$	$4.47E-53$
Average	$7.63E-14$	9.56	$3.35E-27$	0.00	$9.90E01$	$1.54E-29$	$1.94E-27$	$1.95E-27$
STD	$3.36E-13$	$3.65E-15$	$1.50E-26$	0.00	0.00	$6.46E-29$	$8.67E-27$	$5.98E-27$

Table 13: Perfomance of LSRS and SCCSO for dimension 500.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
LSRS								
Best	$-4.3E-19$	$2.9E-39$	$2.12E-11$	0.00	$3.4E-27$	$2.91E-19$	$4.54E-16$	$4.05E-35$
Average	$-4.3E-19$	$2.9E-39$	$4.31E-11$	0.00	$2.61E-11$	$4.08E-19$	$9.0E-16$	$7.96E-35$
STD	0.00	0.00	$1.14E-11$	0.00	$2.32E-11$	$3.62E-20$	$1.52E-16$	$1.95E-35$
SCCSO								
Best	$8.88E-16$	$4.59E01$	$2.52E-54$	0.00	$4.99E02$	$1.15E-55$	$1.27E-49$	$4.56E-46$
Average	$1.42E-15$	$4.59E01$	$2.23E-32$	0.00	$4.99E02$	$3.37E-31$	$6.14E-23$	$5.18E-22$
STD	$1.74E-15$	0.00	$6.86E-32$	0.00	0.00	$1.50E-30$	$2.75E-22$	$2.31E-21$

Table 14: Perfomance of LSRS and SCCSO for dimension 1000.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
LSRS								
Best	$1.3E-18$	$2.9E-39$	$5.34E-30$	0.00	$6.84E-27$	$9.30E-18$	$7.97E-19$	$3.78E-33$
Average	$1.3E-18$	$2.9E-39$	$1.38E-29$	0.00	$7.41E-27$	$1.12E-17$	$1.25E-18$	$7.35E-33$
STD	$4.8E-33$	0.00	$3.68E-30$	0.00	$1.66E-28$	$7.33E-19$	$2.05E-19$	$1.49E-33$
SCCSO								
Best	$8.88E-16$	$9.13E01$	$2.64E-47$	0.00	$10.0E02$	$1.94E-47$	$3.59E-52$	$3.04E-45$
Average	$2.84E-15$	$9.13E01$	$3.26E-33$	0.00	$10.0E02$	$2.68E-30$	$1.61E-30$	$8.14E-25$
STD	$8.74E-15$	$1.46E-14$	$1.46E-32$	0.00	0.00	$7.87E-30$	$7.17E-30$	$3.62E-24$

Table 15: Perfomance of LSRS and SCCSO for dimension 2000.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
LSRS								
Best	$-4.3E-19$	$2.9E-39$	$9.37E-8$	0.00	$1.40E-26$	$2.41E-17$	$9.97E-34$	$7.58E-31$
Average	$-4.3E-19$	$2.9E-39$	$1.69E-7$	0.00	$1.48E-26$	$3.08E-17$	$2.35E-33$	$1.27E-30$
STD	$9.6E-35$	0.00	$3.42E-8$	0.00	$2.44E-28$	$2.31E-18$	$6.91E-34$	$2.02E-31$
SCCSO								
Best	$8.88E-16$	$1.82E02$	$6.63E-56$	0.00	$2.0E03$	$6.01E-55$	$5.25E-56$	$4.24E-41$
Average	$1.07E-16$	$1.82E02$	$1.92E-29$	0.00	$2.00E03$	$9.85E-28$	$1.19E-30$	$1.29E-22$
STD	$7.44E-16$	$5.83E-14$	$8.32E-29$	0.00	0.00	$4.23E-27$	$5.53E-30$	$5.65E-22$

Table 16: Perfomance of SCCSO for dimension 3000.

	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
Best	$8.88E-16$	$2.73E01$	$3.14E-52$	0.00	$3.0E03$	$2.35E-53$	$1.67E-52$	$4.94E-46$
Average	$3.02E-15$	$2.73E01$	$3.097E-27$	0.00	$3.0E03$	$4.66E-30$	$7.54E-31$	$2.40E-21$
STD	$8.73E-15$	0.00	$1.38E-26$	0.00	0.00	$2.00E-29$	$2.51E-31$	$1.07E-20$

Table 17: Average perfomance of LSRS and SCCSO.

Algorithm	Ackley	Levy	Quadric	Rastrigin	Rosenbrock	Schwefel	Sphere	Sum Square
Dim 50								
LSRS	$-6.5E-19$	$2.9E-39$	$2.33E-18$	0.00	$1.38E-18$	$1.91E-11$	$1.38E-18$	$1.42E-18$
SCCSO	$8.88E-16$	5.01	$7.25E-31$	0.00	$4.90E01$	$1.02E-28$	$2.62E-28$	$4.54E-28$
Dim 100								
LSRS	$-6.5E-19$	$2.9E-39$	$1.15E-15$	0.00	$6.94E-16$	$3.98E-10$	$6.94E-16$	$6.98E-16$
SCCSO	$7.63E-14$	9.56	$3.35E-27$	0.00	$9.90E01$	$1.54E-29$	$1.94E-27$	$1.95E-27$
Dim 500								
LSRS	$-4.3E-19$	$2.9E-39$	$4.31E-11$	0.00	$2.61E-11$	$4.08E-19$	$9.0E-16$	$7.96E-35$
SCCSO	$1.42E-15$	$4.59E01$	$2.23E-32$	0.00	$4.99E02$	$3.37E-31$	$6.14E-23$	$5.18E-22$
Dim 1000								
LSRS	$1.3E-18$	$2.9E-39$	$1.38E-29$	0.00	$7.41E-27$	$1.12E-17$	$1.25E-18$	$7.35E-33$
SCCSO	$2.84E-15$	$9.13E01$	$3.26E-33$	0.00	$10.0E02$	$2.68E-30$	$1.61E-30$	$8.14E-25$
Dim 2000								
LSRS	$-4.3E-19$	$2.9E-39$	$1.69E-7$	0.00	$1.48E-26$	$3.08E-17$	$2.35E-33$	$1.27E-30$
SCCSO	$1.07E-16$	$1.82E02$	$1.92E-29$	0.00	$2.00E03$	$9.85E-28$	$1.19E-30$	$1.29E-22$

Table 18: ANOVA test of the average performance of SCCSO and LSRS algorithms on benchmarks.

	Sum of Square	df	Mean Square	F	Sig.
Ackley					
Between groups	.000	1	.000		
Within groups	.000	8	.000	1.182	.309
Total	.000	9			
Quadric					
Between groups	.000	1	.000		
Within groups	.000	8	.000	1.001	.346
Total	.000	9			
Levy					
Between groups	11140.241	1	11140.241		
Within groups	21402.511	8	2675.314	4.164	.076
Total	32542.752	9			
Rastrigin					
Between groups	.000	1	.000		
Within groups	.000	8	.000		
Total	.000	9			
Rosenbrock					
Between groups	1330061	1	1330050.900		
Within groups	2601081	8	325135.150	4.091	.076
Total	3931142	9			
Schwefel					
Between groups	.000	1	.000		
Within groups	.000	8	.000	1.123	.320
Total	.000	9			
Sphere					
Between groups	.000	1	.000		
Within groups	.000	8	.000	2.609	.145
Total	.000	9			
Sum Square					
Between groups	.000	1	.000		
Within groups	.000	8	.000	1.005	.345
Total	.000	9			

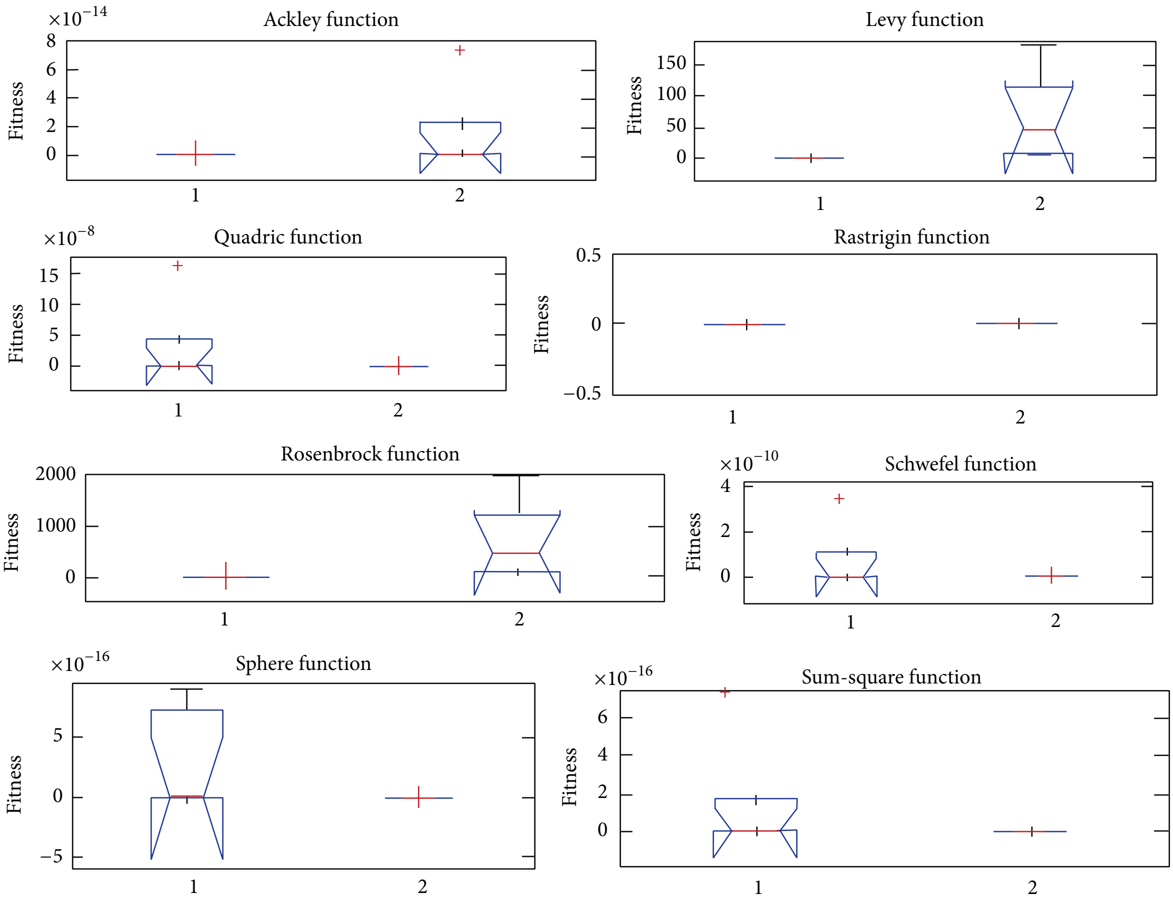

FIGURE 1: A graph showing ANOVA test computed using CSO, MCSO, and SCCSO average performance on test functions. Numbers 1 and 2 on the x-axis denote LSRS and SCCSO, respectively.

SCCSO algorithm has similar performance with the existing LSRS algorithm.

5. Conclusion

The effect of stochastic constriction on characteristics of cockroach swarm optimization algorithm is shown in this paper. Simulation results revealed that the algorithm has good convergence capability. Constriction factor enables the algorithm to maintain swarm stability and enhances local and global searches which resulted in improved convergence and speed of the algorithm.

The proposed algorithm runs fast, solving benchmark problems up to 3000 dimensions; without modifying the algorithm, it can evaluate higher number of variables above 3000 dimensions. Comparisons results of SCCSO with the existing CSO and MCSO show its superiority. Comparison of SCCSO with LSRS shows its ability to compete with known global optimization technique. SCCSO algorithm application to discrete problems will be investigated in further research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This paper is supported by the college of Agriculture, Engineering and Science, University of KwaZulu-Natal through research grant.

References

[1] T. C. Havens, C. J. Spain, N. G. Salmon, and J. M. Keller, "Roach infestation optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '08)*, pp. 1–7, September 2008.

[2] C. ZhaoHui and T. HaiYan, "Cockroach swarm optimization," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET '10)*, vol. 6, pp. 652–655, April 2010.

[3] C. ZhaoHui, "A modified cockroach swarm optimization," *Energy Procedia*, vol. 11, pp. 4–9, 2011.

[4] B. Jiao, Z. Lian, and X. Gu, "A dynamic inertia weight particle swarm optimization algorithm," *Chaos, Solitons and Fractals*, vol. 37, no. 3, pp. 698–705, 2008.

[5] J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George, "Parallel global optimization with the particle swarm algorithm," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 13, pp. 2296–2315, 2004.

[6] C. Grosan and A. Abraham, "A novel global optimization technique for high dimensional functions," *International Journal of Intelligent Systems*, vol. 24, no. 4, pp. 421–440, 2009.

[7] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *Journal of Global Optimization*, vol. 31, no. 4, pp. 635–672, 2005.

[8] L. cheng, Z. Wang, S. Yanhong, and A. Guo, "Cockroach swarm optimization algorithm for TSP," *Advanced Engineering Forum*, vol. 1, pp. 226–229, 2011.

[9] C. ZhaoHui and T. HaiYan, "Cockroach swarm optimization for vehicle routing problems," *SciVerse Science Direct, Energy Procedia*, vol. 13, pp. 30–35, 2011.

[10] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[11] Y. Shi, "Particle swarm optimization," *IEEE Connections*, vol. 1, pp. 8–13, 2004.

[12] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarmoptimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 84–88, July 2000.

[13] C. Yan, B. Guo, and X. Wu, "Empirical study of the inertia weight particle swarm optimization with constrained factor," *International Journal of Soft Computing and Software engineer-Ing*, vol. 2, no. 2, 2012.

[14] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC '05)*, pp. 1777–1784, September 2005.